

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)



Entra en Adictos a través de

E-mail

Contraseña

[Deseo registrarme](#)
[Olvidé mi contraseña](#)

» Estás en: [Inicio](#) [Tutoriales](#) Invocar a un servicio REST securizado, con el soporte de plantillas Spring.



Jose Manuel Sánchez Suárez

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE



[Ver todos los tutoriales del autor](#)



Fecha de publicación del tutorial: 2009-02-26

Tutorial visitado 4 veces [Descargar en PDF](#)

Invocar a un servicio REST securizado, con el soporte de plantillas Spring.

0. Índice de contenidos.

- 1. Introducción.
- 2. Entorno.
- 3. Configuración.
- 4. Invocación.
- 5. Referencias.
- 6. Conclusiones.

1. Introducción

En este tutorial vamos a exponer cómo realizar una invocación a un servicio REST con el soporte de plantillas de Spring y cómo configurar el cliente http para que la invocación se realice con un usuario y contraseña de paso.

El escenario es el de un servicio REST securizado como el que proponíamos en el tutorial de [múltiples contextos de seguridad con Spring Security](#) y un cliente que realiza invocaciones a uno de esos servicios.

Disponiendo de una versión 3.1 de Spring podemos hacer uso del soporte de plantillas para realizar la invocación y para la configuración de autenticación podemos apoyarnos en la librería de Apache commons-httpclient.

2. Entorno.

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 15' (2.4 GHz Intel Core i7, 8GB DDR3 SDRAM).
- Sistema Operativo: Mac OS X Lion 10.7.4
- Spring Security 3.1.0.

3. Configuración.

Suponiendo que hacemos uso de maven, esta sería la configuración de dependencias de nuestro pom.xml:

```

1 <dependency>
2   <groupId>commons-httpclient</groupId>
3   <artifactId>commons-httpclient</artifactId>
4   <version>3.1</version>
5 </dependency>
6 <dependency>
7   <groupId>org.springframework</groupId>
8   <artifactId>spring-web</artifactId>
9   <version>3.1.0.RELEASE</version>
10 </dependency>
```

En el contenido de nuestro applicationContext.xml de spring, la configuración de la autenticación, el cliente http, la factoría de peticiones http y la plantilla que nos da soporte para REST sería la siguiente:

```

1 <context:property-placeholder location="classpath*:app.properties" />
2
3 <bean id="frontCredentials"
4   class="org.apache.commons.httpclient.UsernamePasswordCredentials">
```

Catálogo de servicios Autentia



Síguenos a través de:



Últimas Noticias

» [Orientación a objetos y la importancia del "Tell, Don't Ask"](#)

» [Autentia patrocina al Club KiteSurf Centro](#)

» [Autentia patrocina el I Torneo Voley Playa Terrakas](#)

» [Autentia colabora con la ONG Proyecto Ciclista Solidario](#)

» [Curso de Kanban Core en Madrid con Masa K. Maeda](#)

[Histórico de noticias](#)

Últimos Tutoriales

» [Indexación de documentos en Solr con el soporte de Talend.](#)

» [Configurar múltiples contextos de seguridad en Spring Security 3.1.](#)

» [Transiciones y animaciones con CSS3](#)

» [Configuración y tuning de servidores de producción](#)

» [Revisión de jBPM5](#)

Últimos Tutoriales del

Impulsores Comunidad ¿Ayuda?

0 personas han traído clicks a esta página

sin clicks + + + + + + +

powered by [karmacacy](#)

```

5     <constructor-arg value="{security.front.ws.userName}" />
6     <constructor-arg value="{security.front.ws.password}" />
7 </bean>
8
9 <bean id="frontHttpState" class="com.autentia.tutoriales.ws.rest.CustomHttpState">
10   <property name="credentials" ref="frontCredentials" />
11 </bean>
12
13 <bean id="frontHttpClient" class="org.apache.commons.httpclient.HttpClient">
14   <property name="state" ref="frontHttpState" />
15 </bean>
16
17 <bean id="frontHttpClientFactory"
18   class="org.springframework.http.client.CommonsClientHttpRequestFactory">
19   <constructor-arg ref="frontHttpClient" />
20 </bean>
21
22
23 <bean id="frontRestTemplate" class="org.springframework.web.client.RestTemplate">
24   <constructor-arg ref="frontHttpClientFactory" />
25 </bean>
26

```

El servicio frontRestTemplate nos va a permitir invocar a servicios REST, a través de una factoría que usa el cliente http al que se inyectan las credenciales configuradas mediante el servicio UsernamePasswordCredential y encapsuladas en un servicio propio CustomHttpState. El usuario y contraseña se obtienen de un fichero de propiedades leído a través de un property-placeholder.

El código del servicio que encapsula las credenciales no es más que una clase que extiende de HttpState:

```

1 package com.autentia.tutoriales.ws.rest;
2
3 import org.apache.commons.httpclient.Credentials;
4 import org.apache.commons.httpclient.HttpState;
5 import org.apache.commons.httpclient.auth.AuthScope;
6
7 public class CustomHttpState extends HttpState {
8
9     /**
10     * Set credentials property.
11     *
12     * @param credentials
13     * @see #setCredentials(org.apache.commons.httpclient.auth.AuthScope, org.apache.commons.http
14     */
15     public void setCredentials(final Credentials credentials) {
16         super.setCredentials(AuthScope.ANY, credentials);
17     }
18
19 }

```

La necesidad de esta clase viene dada porque la clase HttpState no está pensada para ser un POJO y el método setCredentials recibe dos parámetros.

4. Invocación.

Una vez disponemos del servicio configurado, podemos inyectar el mismo en cualquier bean manejado por spring e invocar al servicio REST de la siguiente forma:

```

1 import java.io.Serializable;
2 import javax.annotation.Resource;
3
4 import org.springframework.stereotype.Service;
5 import org.springframework.web.client.RestTemplate;
6
7 @Service
8 public class CacheRemoteManager implements Serializable {
9
10     @Resource(name="frontRestTemplate")
11     private RestTemplate restTemplate;
12
13     public String[] getRegionsAvailable() {
14         return restTemplate.getForObject("https://mi-comercio-online/ws/cache/names"), String[].class;
15     }
16
17 }

```

El servicio REST podría ser un ejemplo de recuperación de todas las regiones de cache definidas para el frontal.

Hacer uso de la plantilla es muy sencillo, no tenemos más que establecer la url del servicio e indicar el tipo de objeto en el que la plantilla intentará mapear el resultado de la invocación al mismo. Por defecto, se usa JSON como protocolo de transporte, pero para nosotros es transparente, porque la plantilla se encarga de des-serializar la respuesta al tipo de objeto que le indiquemos.

Del lado del servicio invocado o a invocar tendríamos una configuración similar a la siguiente, con el soporte de Spring MVC:

```

1 @Controller
2 @RequestMapping("/cache")
3 public class CacheResource {
4
5     @Autowired
6     public CacheResource(CacheLocalManager cacheManager) {
7         this.cacheManager = cacheManager;
8     }
9

```

Autor

- » Indexación de documentos en Solr con el soporte de Talend.
- » Configurar múltiples contextos de seguridad en Spring Security 3.1.
- » Uso de componentes JSF de gráficos con el soporte de Primefaces.
- » Uso de un componente JSF de subida de ficheros al servidor con el soporte de Primefaces.
- » Creación de una anotación de validación personalizada para Bean Validator.

Categorías del Tutorial

Spring

Últimas ofertas de empleo

- 2011-09-08 Comercial - Ventas - MADRID.
- 2011-09-03 Comercial - Ventas - VALENCIA.
- 2011-08-19 Comercial - Compras - ALICANTE.
- 2011-07-12 Otras Sin catalogar - MADRID.
- 2011-07-06 Otras Sin catalogar - LUGO.

Jose Manuel Sánchez
sanchezsuarezj

sanchezsuarezj @talend stay tuned!, shortly we will publish more contents about #talend, #TOS and #ETL in @adictosaltrabaj, thanks!!!
21 hours ago · reply · retweet · favorite

sanchezsuarezj Indexación de documentos en #solr con el soporte de #talend. - kcy.me/9zyu vía @adictosaltrabaj
22 hours ago · reply · retweet · favorite

alejandropgarci Orientación a objetos y la importancia del 'Tell, Don't Ask' - kcy.me/9z4f
2 days ago · reply · retweet · favorite

Anais5 Quieres comprender estafa #Bankia? lo explica Juan Moreno,abogado ha conseguido que se admita la demanda #Bankia

Join the conversation

```
9  
10     @RequestMapping(value = "/names", method = RequestMethod.GET)  
11     public @ResponseBody Collection<String> getNames() {  
12         return cacheManager.getNames();  
13     }  
14 }
```

Para más información, me remito al tutorial de [Spring + REST + JSON = SOAUI](#)

5. Referencias.

- <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=springRestJson>
- <http://aruld.info/resttemplate-the-spring-way-of-accessing-restful-services/>
- <http://stackoverflow.com/questions/4615039/spring-security-authentication-using-resttemplate>

6. Conclusiones.

El uso de plantillas en Spring es extremadamente sencillo y potente, si además nos apoyamos en las librerías de Apache Commons, habremos hecho una apuesta segura.

Un saludo.

Jose

jmsanchez@autentia.com

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

Por favor, vota +1 o compártelo si te pareció interesante

Share |

0

Animáte y coméntanos lo que pienses sobre este **TUTORIAL**:

» [Regístrate](#) y accede a esta y otras ventajas «



Esta obra está licenciada bajo [licencia Creative Commons](#) de Reconocimiento-No comercial-Sin obras derivadas 2.5

Copyright 2003-2012 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

